MODELING OF PROBLEM

RECIPE.

SYMBOLIC-GOLDEN SOLUTION.

FRANK K. N. APPIAH

CONTENTS AT GLANCE

Preface.....6

INTRODUCTION.....9

CHAPTER 1 AN ALGORITHMIC BEAUTY OF PROBLEM RECIPE.....13

1.0 ABSTRACT.....14
1.1 PROBLEM RECIPE.....15
1.2 ALGORITHMS OF PROBLEM RECIPE 17
1.3 CLAIMS.....20
1.4 DRAWING.....22

CHAPTER 2 APPLICATION OF GOLDEN-MAGIC SOLUTION.....25

2.0 ABSTRACT.....26
2.1 DESCRIPTION.....26
2.2 BACKGROUND OF APPLICATION.....28
2.3 SUMMARY OF APPLICATION.....37
2.4 BEST MODES OF CARRYING OUT APPLICATION40

CHAPTER 3 SYMBOLIC DECOMPOSITION MODELING.....42

3.0 INTRODUCTION.....43
3.1 MODELING OF PROBLEM RECIPE.....43
3.1.1 CLASS DIAGRAMS.....47
3.1.2 USE CASES.....50
3.1.3 SEQUENCE DIAGRAM.....51
3.2 SOFTWARE MODELING.....51

CHAPTER 4 UML METAMODEL.....54

4.0 Introduction.....55
4.1 UML Model.....55

GLOSSARY.....61

Further Reading.....67

Bibliography.....70

Appendix A: Computer Codes.....72

Appendix B: UML Metamodels in Symbiolic Decomposition. 77

Preface

This book is about symbiolic decomposition modeling using methods of software research. The concept of symbiolic decomposition is the chosen research question. The review of symbiolic decomposition is a conduct of literature only.

Illustration Index

Illustration 1: Golden-Magic Solution Diagram.....22

Illustration 2: Hand Diagram.....23

Illustration 3: Activity Diagram.....44

Illustration 4: State Diagram.....45

Illustration 5: Main Class Diagram.....47

Illustration 6: SymbolicSoln Class Diagram.....48

Illustration 7: ProblemSoln Class Diagram.....49

Illustration 8: Decomposition Class Diagram.....49

Illustration 9: Solver Use Cases.....50

Illustration 10: Sequence Diagram.....51

Illustration 11: UML Schema Main-Tags.....56

Illustration 12: Header Tag.....56

Illustration 13: Content Tags.....57

Illustration 14: Model Tags.....59

Illustration 15: Package Tag.....60

INTRODUCTION

The developing of hypothesis is based on the problem recipe of symbiolic decomposition by algorithmic notations. This problem recipe is an attempt to model verbally a solution to universal problems. Problem recipe is a process of conception to represent problems, solutions, parties and recipes. The study of the hypothesis on problem recipe of symbiolic decomposition is the present application of golden-magic solution. In the conduct of hypothesis and its study is a sofware research composed of algorithm design, programs and unified modeling language (UML Diagrams and UML Metamodels). The software research is termed as modeling of symbiolic decomposition. The computer experiment on the conduct of hypothesis and it's study is the 3 computer programs in the Appendix. The computer experiment is a group of Decomposition class, ProblemSoln class and SymbiolicSoln class.

In execution will be a population of problem recipes composed of instances of Decomposition, ProblemSoln and SymbiolicSoln. The significant difference is the allowance to conclude on the effects of existence of problem recipe populations to be able to reject a null hypothesis in a significant solution test. The sample meaning of problem recipes in a population sample is a static process. The statistical tests for effects are based on the ratio of of the size of the difference in sample means and sampling error. The inferential test is estimated from measures of variability in the sample. An inferential test on *SymbiolicSoln* of problem recipes will be based on the following variables:

- *partysolver*,
- *recipetype*,
- *nature*,
- *solvertype* and
- *recipe*.

An inferential test on *ProblemSoln* of problem recipes will be based on the following variables:

- *problemType,*
- *solutiontype,*
- *pemType* and
- *slnType*.

An inferential test on *Decomposition* of problem recipes will be based on the following variables:

- *symbol ,*
- *problem*,
- *settlement*,
- *settlecount* and
- *settleType*.

Variability of the sample indicates the amount of sampling error present because samples with high variability are less likely to be a good representations of the population[Dawn].

CHAPTER 1
AN ALGORITHMIC BEAUTY OF PROBLEM RECIPE.

A SYMBOLIC SOLUTION AND GOLDEN-MAGIC SOLUTION.

1.0 ABSTRACT

An algorithmic beauty of problem recipe for generating a solution is described and sketched. It is simple-minded yet a solution in a natural universe. The ways of representing a problem is also discussed and developed. There is the theorem of “Universal Problem” been discussed and developed. There is also a-mentioned of the “Count Problem Solver” theorem that states that the least number of problem solvers needed for a recipe to a particular problem in the natural universe. There are two limitations to a problem solution. The two main types of problem solution are described in this publication. The sketch of the Symbolic solution is clearly drawn and that also of the Golden-Magic is drawn. In conclusion, problem recipe is a new method of conception and in the nutshell, it is a process of conception to the representation of problems, solutions, parties and recipes on a

single diagram. An universal problem is natural in all sense and requires the solvers to dip-deep into problem with their own hands. A free hand sketch is provided to show the right signals needed as a problem solver in a deep problem space. Problem recipe will allow the solvers not to rush into later hesitations but settle down to a particular recipe.

1.1 PROBLEM RECIPE

A simple-minded and a beautiful problem recipe for generating a solution in a natural universe is constructed and developed here. A problem can be represented either naturally or universally, then a problem recipe is either natural and/or universal.

Theorem (Universal Problem)

In a natural universe, there can only be one problem.

This means that there is only one problem in a natural universe. The nature of origin of the problem is the root of the problem. A positive case or negative case can be the limit to a problem solution. A positive case drives the problem solution to a more settle-down and positive problem recipe whiles the negative case guidelines to a more settle-up and prohibited problem recipe. A positive problem recipe is a permitted and always authorized solver. A negative case opens a general solution to a problem whiles a positive case closes a general solution to a problem. Because of the hesitations of negative cases, a problem solver needs to settle down on a 2-5 count problem recipes.

Theorem (Count Problem Solver)

There is at least 5 problem solvers for a recipe to a particular problem in the natural universe.

With a symbolic solution, there are two *symbol objects* (Check source code in Appendix):

- recipes- magic, rational, golden;
- solver party- positive and negative.

1.2 ALGORITHMS OF PROBLEM RECIPE

In this section, I would describe a couple of algorithmic problem recipes. First to look at is the main algorithm of problem recipe and secondly is the algorithm of universal problem.

Procedure: Given a problem, p in a universe with solutions, sol of 5 solvers. If the p is less than 2 then it will require that the solver find the root of the problem as nature

of problem and start to dig into the problem with the solver's own hands. If a positive case is yield then settle down with the positive solvers. Otherwise, settle up with the negative solvers with the negative cases. That is it, end. Let's look at the algorithm now.

1. ALGORITHM (PROBLEM RECIPE):

```
Given a problem, p in a universe with solutions,
sol of 5 solvers;
    if(p < 2) then
    Find the root of problem and dig into problem
with own hands;
        if(positive case) then
           Settle-down(+sol);  //+solver
        else
         Settle-up(-sol);   //-solver party
        END IF
    END IF.
```

Procedure: Given a problem, p in a natural universe with a problem solver, sol. If the existence of the solvers are within the range of 2 to 5 then print that the problem is the only one in the natural universe. That is the end. Let's look at the algorithm now.

2. ALGORITHM (UNIVERSAL PROBLEM)

```
 Given a problem, p in a natural universe with a
problem solver, sol;
    if ( 2 ≤ sol && sol ≤ 5) then
        print("The Problem is the only one in the
natural universe");
    END if.
```

1.3 CLAIMS

A natural and universal problem recipe described in the DESCRIPTION sheet hands over the following claims:

1 . There is only one problem in a natural universe.

2 . A problem recipe is natural and/or universal.

3 . A recipe for a problem, one(1) in a natural universe will need not less than 2 problem solvers and not more than 5 problem solvers(Ps) ie $2 \leq Ps \leq 5$.

4 . The root of the problem is given in the nature of origin.

5 . There are two limitations to a problem solution ;- positive(s) (+) and negative(s) case(s).

6 . Because of later hesitations, one needs to settle down on the 2-5 count problem recipe.

7 . A golden and magic solution settles in two-problem decomposition.

8 . A golden and magic solution is influenced by the root analysis of 5 problem solvers. (FIG. 2)

9 . A golden and magic solution is lead by one natural or universal solver in addition to the 5 problem solvers in partitions.

10 . With a symbolic solution, then there is an equality of only two symbol objects; ie Three circles in a triangle. (FIG. 1)

11 . The root of the problem is used in the division solution.

12 . (2-5) count problem recipe is invented because of human reasoning that follows:

12.1 There are 2 parts of a folded-hand.

12.2 A folded-hand has 5 finger parts.

12.3 The are 2 directions with a folded-hand. (FIG. 3)

12.4 The folded-hand with a ball in the palm also has 2 parts:

The 4 fingers out-warding from the palm holds the ball(pen) and the 5^{th} finger points upwards. The thumb is the 5^{th} finger.

12.5 A finger of a hand has 3 main parts and 2 lines of the finger part in the 2 mains.

13 . There are 2 forms of the golden and rational decisions for the positive and negative solver parties.

14 . There are only 3 parts of the problem on the focus for golden and rational recipe. (FIG. 1).

1.4 DRAWING

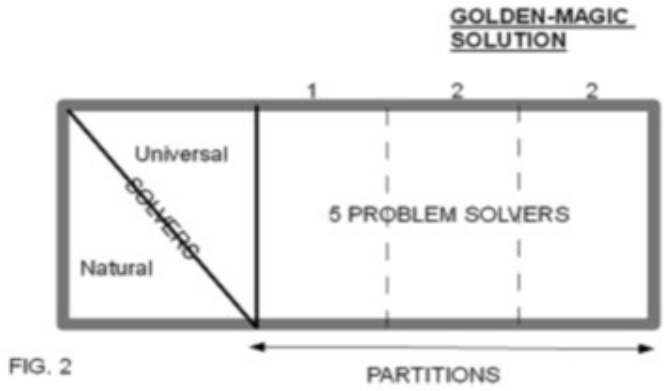

Illustration 1: Golden-Magic Solution Diagram

Source: (c) Frank Appiah. 2016. Golden-Magic Solution Diagram. OpenOffice Drawing Tool. Author's work of art, Patent Regulation.

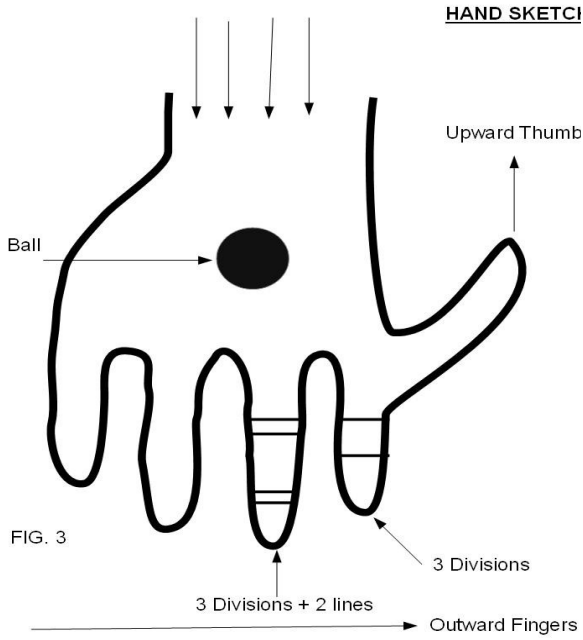

Illustration 2: Hand Diagram

Source: (c) Frank Appiah. 2016. Hand Diagram. OpenOffice Drawing Tool. Author's work of art, Patent Regulation.

CHAPTER 2
APPLICATION OF GOLDEN-MAGIC SOLUTION.

2.0 ABSTRACT

This is the application of golden-magic diagram to a problem decomposition in general. A problem recipe is a method of generating a solution in a natural universe. The nature of problem recipe is a natural or universal. The process of representation of problems, solutions and party solvers involved can be in a single diagram.

2.1 DESCRIPTION

TECHNICAL FIELD : A GOLDEN-MAGIC SOLUTION

The present application relates to a problem recipe as a new method of conception in generating a golden-magic

solution(GMS), and more particularly a process of conception of representation of problems, solutions, parties, recipes in a single diagram [FIGURE 2]. According to this application, the diagram of golden-magic solution is employed.

TECHNICAL FEATURES

LOGIC, TECHNOLOGY, PROCESS OF RESEARCH (PSYCHOLOGY), COMPLEXITY, EXPERIMENT, PROBLEM SOLVING, MEMORY, SOCIETY AND CULTURE.

SUBJECT MATTER

TWO-PROBLEM DECOMPOSITION

DEFINITION:

TWO-PROBLEM DECOMPOSITION IS MADE UP OF TWO SETTLE-SOLUTIONS MAINLY GOLDEN AND MAGIC.

KEYWORDS

MAGIC PROBLEM ;GOLDIC PROBLEM; GOLDEN-MAGIC;

PROBLEM SOLVER;

2.2 BACKGROUND OF APPLICATION

The definition of subject matter as in the background of the application is characterized by as claimed [**CLAIM 7**]. The definition is characterized in that "A golden AND magic solution settles in two-problem decomposition. The color feature of golden draws attention to the very metal called Gold from the Ashanti Region of Ghana. The matter of gold is to set us up into the motion of problem solving with the

metal. In chemistry, the element of gold is Au. Today, we have a union in Africa called AU for the whole of Africa continent. Then a logical decomposition of the term Africa Union can be addressed to the gold element- Au. The elements of metals can be a source of solution to the naming of problems that we experience.

This experience can be the correct thing in the natural environment that may have drawn one to the attention of a formed union or composition. The beauty of gold is the source of beauty to the algorithmic step of elementary two-problem decomposition. The gold matter is used in Ghana in general to create jewelery for the beauty of men and women of the land especially Asantes (plural of Ashanti). The symbols of Adinkra (Ashanti Symbols) are mainly used in the design of the tail of gold jewelery. The gold is a symbolic matter for some people of Ashanti Region.

The number of alphabets of the Au (gold) is 2. This is a logical two-problem decomposition in a golden-magic

solution. Then basing a number of alphabets on a problem decomposition is a solution to most problems. All problems need to be abstracted from the thing itself. Abstraction brings into play the concept of nouns. A naming of thing is a major step to have a beautiful solution and it all boils down to a noun of a thing, place, person etc.

Just name it to have a solution to a problem. Look at natural means in your environment and element to get the major alphabets to select words of noun. There you will have a natural solution from the decomposition. Au is an element of gold and it is a world-established element in chemistry subject. This makes the element Au, a real world thing and it's adoption for describing a group, union, thing, person is universal.

On intellectual fun, Au for a group can be called *augrounp* or augroue, augroun, *augron* or more, that of thing can be termed *authine*, that of a person can be named Kwabenau, *Kwabenaugh*, *Abenau* (female of not Adwoa),

Abenaugh (female of not Adwoa but of Ghana), Adwoau (female of not Adwoa), Adwoaugh (female of not Adwoa but of Ghana) and a street-name of a place can be austreet or autreet or auree or aurret or auname or retause, more. Let us look at the definition of onamastic. It is the study of naming of things, persons or places. On the *augrounp* group, the name was formed from the concatenation of *au* and *group* with an *n* placed in between *u* and *p*.

The second group called augroue is formed from *au* and *group* by replacing the *p* letter with an *e*. The third group is formed from *au* and *group* by replacing *p* with *n*. I will leave the fourth and move to the persons. I will start with Kwabenau formed from the Tuesday-born name *Kwabena* and *u* from the second letter of *au*. The second person name if also formed from Kwabena and ugh from both the second letter of *au* and *gh* to represent Ghana.

The last naming is about places as streets. The first street-name is austreet formed by concatenation au and street.

The second name of street is autreet which is also formed by concatenation *au* and t*reet* of street (without the s). The third name of street is *auree* formed from that same au and ree (without *st* starting and *t* ending). The settlement in two-problem decomposition is to draw attention to the problem recipe that settles in the people's land.

There are a number of recipes sitting on the lands of the people and just needs a subjection of thought on this matter. In addressing the a solution as magic is a method of problem recipe. Magic is to tackle a problem with some kind of supernatural powers living in a kind of medium. There are other ways of magic depicted by the reflection of shadows on a bar of gold holding one's hand.

The reflection of shadows is a sign of magic in the sense that the bar holds the shadow of the holding-man. Somethings need to be reflected upon in order to understand the problem recipe as a solution. Reflection of things sometime result to checking one's mind if there is an

application to such phenomenon. Memory of a man is a needed thing to reflect on past, present and future things that bring solution to our problems. The solution to a problem is a need for a real world solution in the mind's of others out of one's dimension.

There is always a solution that exist out of the realms of one but in the other far away. This is to say that daily real world solution has been used elsewhere unconsciously and so need the attention in memory to look at them again. My own pro-gramme of study, writing and reading basically on subject matters of Information and Communication Technology made me focus more on the attention of memory for past, present and future.

The approach that i took was to experiment on the exact that i have planned in the past and bring to the present as it is needed. The need for my own studies do require of my own in a real existence in a world that i have lived before. The problem solver is one that focus on the attentions of memory

to the problems that needs to be studied. The much-look for solution is in the mind of the problem solver and needs to call as in dialing into one's own mind of existence. There are things in the mind also as it is in the world. The mental world is a container of these things that has existed in a real world that we live now.

The emergence of the mental world and the real world is a place to find the existence of solutions to our major problems. The problem solver needs to emerge of the mental world and name things that he or she sees. The solver then proceed to look for things of the environment that matter to the achieve the such named thing in the real world. Many solvers have laid the ground-terms of solutions to the very dimension of thought from the mind of existence.

The solver will exit a named problem if it is studied in the mind. The existence of mind is a done deal of problems. In the mind as we reflect, we can decompose and assemble a new thing of great value to us and others. Search through the

mind of existence to the problems concerning us in our lands. The focus in the mind is the search to the named problem as thinking.

Thinking to decompose into a major and a minor problem can help us further decompose to the very small level that the problem can attain. The level so small can be the use of large numbers of small transistors in a computer to solve. The decomposition of problems into two solutions can be represented into many other two solutions like the branches of a tree or nodes of graph. The monitor of the computer helps to throw-much light on the representation of problems at display in a moment.

Just like the light from the bars of gold shines, the light from the monitor of the computer can make us see a light through the problem. The concept of decomposition is to find a shine enough to show a clear cyberspace world of the computer. The extension of the memory is a computer memory that helps keep the many bothering issues in the mind of

problems in a real world. The bits of the problem solution is represented in the computer memory for two reasons.

The first is to save our thinking in a word processor to help further in a search for a solution and the second is to compute the search for things in our own thinking. Even for the computers, the data of thinking needs to be existence before computation can begin. A cabbage in, A cabbage out (ACIACO) defaulter can create more problems in finding a better solution to our problems.

An ACIACO defaulter is a two-problem de-composition method to problem solving in real world. This is because to check the correct functioning of a machine processing a two-problem decomposition solution one needs to pass ACIACO defaulter. An ACIACO defaulter will always bring the same cabbage in as an output to a problem in computation. It is an open-system which takes in some corrupt data and brings out the same corrupt data to the output of the display after computation. It can be equalized as x=x.

The characterizing portion of the equation is x is equal to x. It can be used to check if the same results of a data is resolved after moving through the wires of the computer.

The settler of two-problem decomposition is in a magic and golden solution. Many bars of gold in a room can light under dark environment and this light can be tunnel to closed spaces of an environment to create a yellowish/goldish color in a room. The gold is characterized by people in the world as an influencer of status in world economy for the Ashanti people. The golden-magic recipe of solution is an influencer of status in solving magic problem and goldic problem.

2.3 SUMMARY OF APPLICATION

The present application provides a golden-magic methods of solution as a problem recipe for goldic and magic problems.

The color feature of gold is used to draw antecedents for reasoning about goldic and magic problems in this application. The element of gold as Au is used in constructing a de-composition to assemble new meanings to the nature of gold. This is a means to make the naming of objects in one's mind natural and universal. The goldic reason of two in the problem decomposition is mainly due to the golden and magic. The second reason is attributed to the number of alphabets in Au element which is two. All problems need to be abstracted from the thing itself and just naming it gives a solution to the problem. We took a look at natural means in your environment and an element to get the major alphabets to select words of noun. Gold is a symbolic matter for some people of Ashanti Region. The presentation of thinking is to decompose a problem into a major problem and a minor problem can help us further decompose to the very small level that the problem can attain. The approach to problem solving is to experiment on the exact that have been planned in the past and bring to

the present as it is needed. The finding of existence of solution in a real world it to emerge into the mental world. Somethings need to be reflected upon in order to understand the problem recipe as a solution. Reflection of things is to check one's mind if there is an application to a such phenomenon. The memory of a man is a needed thing to reflect on past, present and future things that bring solution to our problems. A solution exist in a dimension different from the one searching for. The two needs to merge for a particular solution to emerge from the mental dimension of our mind.

CUMMULATIVE TO CLAIMS

CLAIM 7: A golden and magic solution settles in two-problem decomposition.

SUMMARY OF DRAWING

Figure 2 is an illustration of a diagram of golden-magic solution.

2.4 BEST MODES OF CARRYING OUT APPLICATION

In order to facilitate an understanding of the present application, the following definitions which are used herein are presented:- Check the Glossary page.

CHAPTER 3
SYMBOLIC DECOMPOSITION MODELING.

3.0 INTRODUCTION

This chapter is about the modeling of problem recipe specifically, symbiolic decomposition specification. The graphical form of model diagrams from Umbrello UML Modeller is depicted. These model diagram includes state diagram, activity diagram and class diagrams. Lets see the details of the model diagrams in the symbolic decomposition modeling.

3.1 MODELING OF PROBLEM RECIPE

In modeling the problem recipe by symbolic decomposition, the activities of the process are modelled by activity diagrams. Activity diagram is the name of the graphical form of modeling in UML schema. The diagram for this specific concept is as shown below:

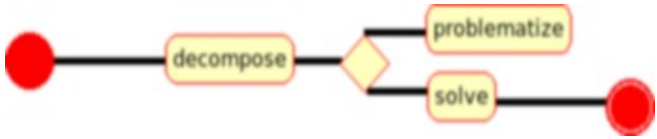

Illustration 3: Activity Diagram

Source: (c) Frank Appiah. 2018. Activity Diagram. Umbrello UML Modeller. Author's work of art.

The activity diagram uses an activity name called decompose as the symbolic decomposition, problematize as the problem and solve, activity name as the solution. The decompose activity is a join activity of problematize and solve activities. The red circle is the entry into the activity and it is a start activity. Ofcourse, red symbolizes a problem or a stop activity. The second circle has two circumference to rather indicate stop activity. In capturing the activity, I read the diagram as :

1. On a problem, start (first red cirlce) an activity of symbolic decomposition.
2. On decompose activity, join the activities of problematize and solve.
3. On fork joins, take out problems and solve until you stop on a solution.

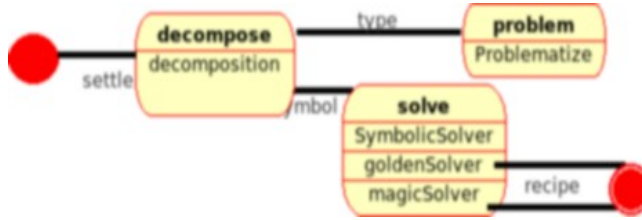

Illustration 4: State Diagram

Source: (c) Frank Appiah. 2018. State Diagram. Umbrello UML Modeller. Author's work of art.

The modeling of states of the symbolic decomposition in graphical form is as shown above. A state is a name of model state of decomposition process depicted as a state diagram. Each state has a name and several activities in that state of a decomposition model. The *statetype* shows the type of state active at a point in time (See Chapter 4).

The *settle* state is of type 2 and has no activities to act upon. The *decompose* state has only one activity and that is decomposition. After the decomposition state is the both the *type* state and *mbol*(symbol) state. The type state leads to the *problem* state, which has an activity of problematize. The mbol state has a lead to the *solve* state that has three activities

in its details namely – SymbolicSover, goldensolver and magicsolver. The solve state creates a lead for a *recipe* state. At the recipe state, there is three leads to a stop state from each activity. With the state diagram, I will read as:

1. At a *start* state of a decomposition process model, move to a settle state.
2. At a *settle* stateline, points to a decompose state.
3. At a *decompose* state, create a decomposition activity in the process and move to a type state.
4. At *type* stateline, points to problems state.
5. At *problem* state, create a problematize activity and move back to decompose state.
6. From a decompose state is a symbol state line.
7. At *symbol* stateline, points to the solve state.
8. At *solve* state, creates three specific activities namely *SymbolicSolver*, *goldenSolver* and *magicSolver* and move to a settle state line.
9. At *settle* state line, points to a stop state to end activity.

3.1.1 CLASS DIAGRAMS

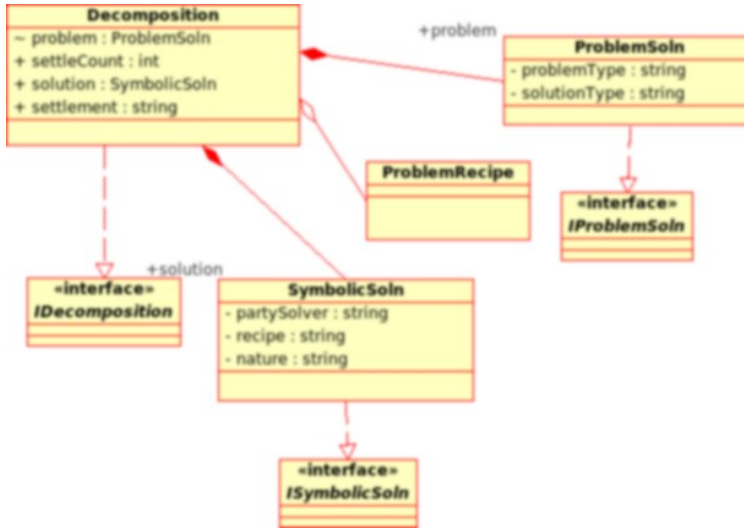

Illustration 5: Main Class Diagram

Source: (c) Frank Appiah. 2018. Main Class Diagram. Umbrello UML Modeller. Author's work of art.

There is a *ProblemRecipe* package containing three classes and interfaces. The three classes in the package are:

1. Decomposition
2. SymbolicSoln and
3. ProblemSoln.

The three interfaces in the package are:

4. IDecomposition
5. ISymbolicSoln and

6. IProblemSoln.

The methods of Decomposition class are:

1. getProblem();
2. setProblem(problem : ProblemSoln);
3. getSettleCount();
4. setSettleCount(int settlecount);
5. getSolution();
6. setSolution(solution: SymbiolicSoln);
7. getSettlement();
8. setSettlement(settlement: String).

A class diagram is the name of a model diagram in the class specification of variables and methods of the decomposition process. See the detailed class diagrams below.

SymbolicSoln Class

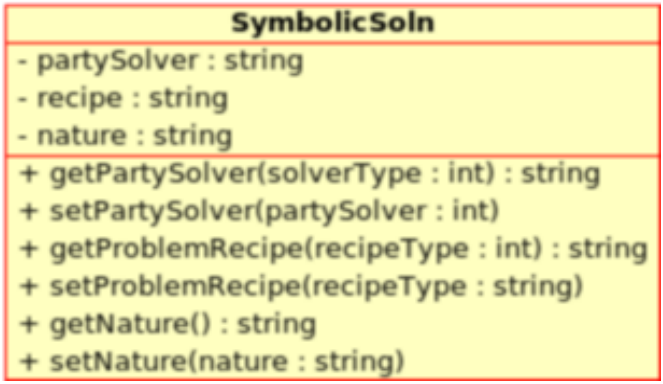
SymbolicSoln
- partySolver : string
- recipe : string
- nature : string
+ getPartySolver(solverType : int) : string
+ setPartySolver(partySolver : int)
+ getProblemRecipe(recipeType : int) : string
+ setProblemRecipe(recipeType : string)
+ getNature() : string
+ setNature(nature : string)

Illustration 6: SymbolicSoln Class Diagram

Source: (c) Frank Appiah. 2018. SymbolicSoln Class Diagram. Umbrello UML Modeller. Author's work of art.

ProblemSoln Class Diagram

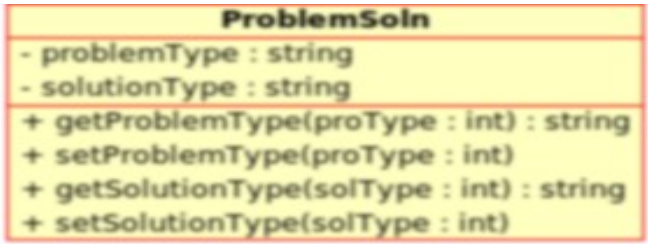
ProblemSoln
- problemType : string
- solutionType : string
+ getProblemType(proType : int) : string
+ setProblemType(proType : int)
+ getSolutionType(solType : int) : string
+ setSolutionType(solType : int)

Illustration 7: ProblemSoln Class Diagram

Source: (c) Frank Appiah. 2018. ProblemSoln Class Diagram. Umbrello UML Modeller. Author's work of art.

Decomposition Class Diagram

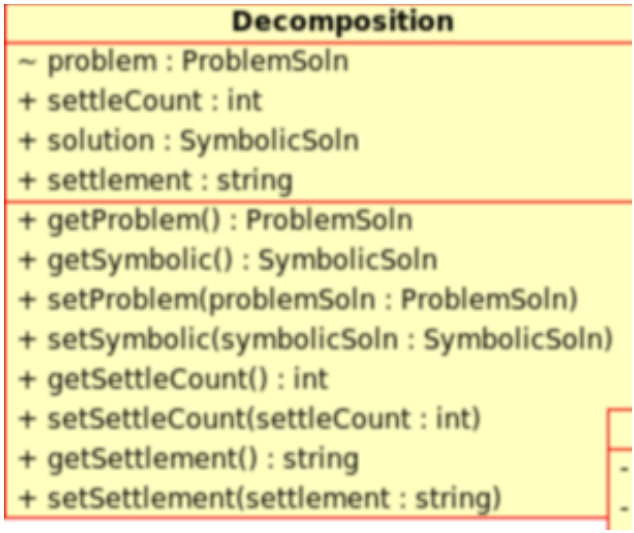

Illustration 8: Decomposition Class Diagram

Source: (c) Frank Appiah. 2018. Decomposition Class Diagram. Umbrello UML Modeller. Author's work of art.

3.1.2 USE CASES

Use case diagram is the name of the model diagram in the decomposition process. There are four use cases in the decomposition processs namely:

1. Decomposition_case
2. Problem_case
3. Solution_case
4. Settlement

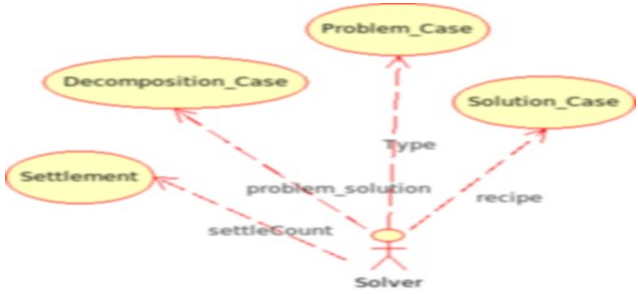

Illustration 9: Solver Use Cases

Source: (c) Frank Appiah. 2018. Solver Use Cases. Umbrello UML Modeller. Author's work of art.

See the use case diagram above.

3.1.3 SEQUENCE DIAGRAM

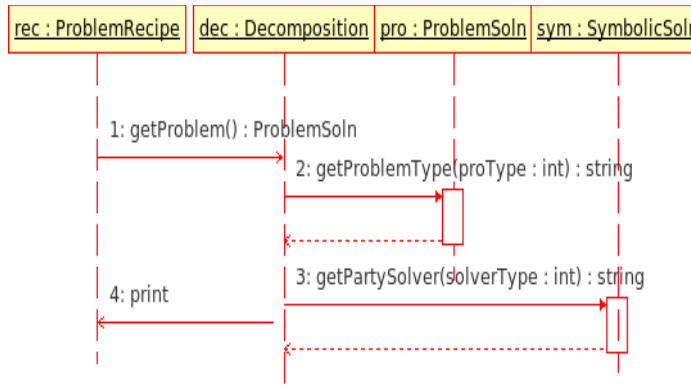

Illustration 10: Sequence Diagram

Source: (c) Frank Appiah. 2018. Sequence Diagram. Umbrello UML Modeller. Author's work of art.

3.2 SOFTWARE MODELING

This is about the Java classes developed from the class diagrams as described above. The Java classes for Decomposition class, ProblemSoln class and SymbiolicSoln class are listed as computer codes in Appendix A.

CHAPTER 4
UML METAMODEL

4.0 Introduction

UML metamodeling is about the process of UML diagramming. In the process of creating state diagram, sequence diagram, activity diagram and class diagrams; a business object in XML format is generated as a metamodel of model diagrams.

The metamodel is a scheme of extensible markup languages objects. A metamodel is a model about model.

4.1 UML Model

UML model is a metamodel. UML model is a model about stereotype models. There are three stereotypes in UML model namely:

1. Datatype
2. Folder and
3. Interface.

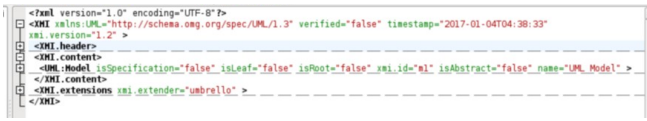

Illustration 11: UML Schema Main-Tags

The default schema from OMG shows three main tags in the XMI structure namely (See Figure 11):

- Header
- Content (<XMI:content></XMI:content>)
- Model <XMI:model></XMI:model>)

The model tag is visible within the content tag. The header tag is visible out of the content tag.

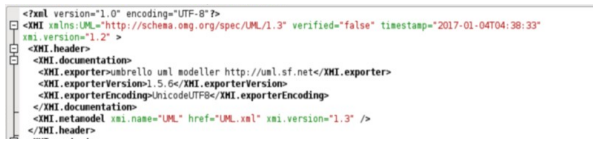

Illustration 12: Header Tag

The Figure 12 is the details of the header with its sub-tags. There are about 5 new tags to consider in the header :

1. <XMI:documentation>
2. <XMI:exporter>

3. <XMI:exporterVersion>
4. <XMI:exporterEncoding>
5. <XMI:metamodel>

The content tag is made up of :

1. model tags: <UML:model>
2. stereotype tags: <UML:stereotype>

Illustration 13: Content Tags

The model has attributes like *isSpecification*, *isLeaf*, *isRoot*, *isAbstract, stereotype, id* and *name*. The stereotype tag has attributes like *isSpecification*, *isLeaf*, *visibility*, *namespace*, *is Root*, *isAbstract* and *name*. The model identifier includes logical view, usecase view, component view, deployment view and entity relationship model.

The model tag is made up of (See Figure 14):

1. package tag: <UML:package>
2. class tag: <UML:class>

3. interface tag: <UML:interface>
4. abstraction tag: <UML:abstraction>
5. association tag: <UML:association>
6. extension: <XMI:entension>

Illustration 14: Model Tags

The package tag is made up of (See Figure 15):

1. datatype: <UML:datatype> - There can be several datatypes in a package. The datatypes include:
   bool, string, int, short, double, float, long, unsigned int, unsigned short and unsigned long.

   Name attribute has values like one of the listed datatypes above. Datatype has attributes like stereotype, isSpecification, isLeaf, visibility, namespace, isAbstract and id.

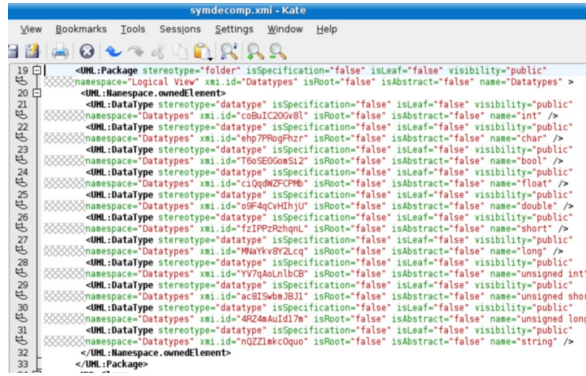

Illustration 15: Package Tag

The complete metamodel of Symbiolic Decomposition UML is in Appendix B.

GLOSSARY

- ✓ **Problem recipe**: a process of conception of representation of problems, solutions, parties, recipes in a single diagram.
- ✓ **Golden-magic solution** : problem recipe in a single diagram.
- ✓ **Magic problem**: problem of its nature in solving involves magic.
- ✓ **Goldic problem:** problem in solving it involves golden nature of thought recipe.
- ✓ **Two-problem decomposition**: made up of two settle-solution mainly magic and golden.
- ✓ **Magic:** tackle a problem with some kind of supernatural powers living in a kind of medium.
- ✓ **Reflection of things**: result to checking one's mind if there is an application to such phenomenon.

SOFTWARE RESEARCH

- ✓ **Concept of decomposition** :search for a shine enough to show a clear cyberspace world of the computer.
- ✓ **Memory of a man**: needed thing to reflect on past, present and future things that bring solution to our problems.
- ✓ **Magic solution**: method of problem recipe.
- ✓ **Extension of memory**: computer memory that helps keep the many bothering issues in the mind of problems in a real world.
- ✓ **ACIACO defaulter**: two-problem decomposition method to problem solving in real world.
- ✓ **ACIACO defaulter**: open-system which takes in some corrupt data and brings out the same corrupt data to the output of the display after computation.
- ✓ **Simple-mind**: the thought of decomposition of things and supplication of assembly of new great things.
- ✓ **Mental world**: a container of these things that has existed in a real world that we live in now.

Index

abstraction....58
Abstraction....30
ACIACO....36
ACIACO defaulter....36, 62
activity diagram....43, 44, 55
application....26, 27, 28, 32, 37, 38, 39, 40, 61
association....58
attention....33
attribute....59
business....55
class....43, 48, 51, 55, 57, 73, 74, 76
class diagrams....43, 48, 51, 55
component view....57
computer....35, 36, 51, 62
computer codes....51
computer memory....36, 62
conception....26
container....34, 62
content tag....56, 57
Count....14
Datatype....55, 59
decomposition....26
Decomposition class....48, 51
decomposition process....45, 46, 48, 50
defaulter....36
deployment view....57
dimension....34
division solution....21
element....29
emergence....34
entity relationship model....57
environment....34
existence....34
experiment....33

extension....35, 58
Extension....62
Extension of memory....62
Folder....55
gold....29
golden....17, 22
Golden-Magic....14
golden-magic solution....29
group....30
hand sketch....15
header....56
header tag....56
human reasoning....21
influencer....37
interface....58
Interface....55
Java classes....51
limit....16
logical decomposition....29
logical view....57
magic....17
Magic solution....62
major....38
Memory....33
mental....34
metamodel....55, 57, 60
method....14
minor....38
model....43, 45, 46, 48, 50, 55, 56, 57
model diagrams....43, 55
model tag....56, 57
motion....28
naming....29
natural....14
nature....16, 17, 20, 26, 61, 70, 73, 74
negative....17

negative case....16
nouns....30
onamastic....31
one problem....16
open-system....62
package....47, 57, 59
phenomenon....33, 39, 61
positive....17
positive case....16
problem 14, 15, 16, 17, 18, 19, 20, 21, 26, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 43, 44, 45, 46, 48, 61, 62, 68, 69, 76, 77
problem recipe…14, 15, 16, 17, 20, 21, 26, 32, 37, 39, 43, 61, 62, 68, 69
Problem Solver....14
problem solving....36, 38, 62, 68
problem space....15
ProblemSoln class....51
process....14, 26, 27, 43, 45, 46, 48, 50, 55, 61
rational....17
rational decisions....22
rational recipe....22
realms....33
reasoning....38
recipe….14, 15, 16, 17, 20, 21, 22, 26, 32, 37, 39, 43, 46, 61, 62, 68, 69, 73, 74
reflection of shadows....32
root....16
root analysis....21
schema....43, 56, 71
search....35, 36, 62
Search....34
sequence diagram....55
settle-down....16
settle-up....16
settler....37
signals....15

single diagram....14
solution.14, 15, 16, 17, 20, 21, 26, 27, 28, 29, 30, 32, 33, 34, 36, 37, 38, 39, 44, 48, 61, 62
solver party....17
state diagram....43, 45, 46, 55
stereotype....55, 57, 59
structure....56
supernatural....32
SymbiolicSoln class....51
symbol objects....17
Symbolic....14
symbolic solution....21
Symbolic solution....14
thinking....35
thought recipe....61
two-problem decomposition....20
UML....43, 54, 55, 57, 58, 59, 60, 71
UML diagramming....55
UML metamodeling....55
Universal Problem....14
universe....14
usecase view....57
world....30, 33, 34, 35, 36, 37, 39, 62, 69, 70
XML format....55

Further Reading.

Scientific Methods. The methods of research in the scientific sense is conducted on the understanding of the assumptions that come along with the methods of science. Thinking about the field of Biology can help one to understand how influential research is in problem recipe or problem solving. In the biological research, problem recipe did investigate the way the hand of the human body is applied to the conception of problem recipe. The knowledge of the hand helps in this research by determining that :

1. The hands are two on the human body.
2. There are five fingers on each hand.
3. There are three divisions in each finger and more.

We need to keep in mind these assumptions when solving a problem with the conception of problem recipe. Since the earlier times, scientific study requires observations.

Observation is really what sets scientific fields apart from other fields of study[Dawn]. The problems we want to solve must be learned in its happenings and in the world around us.

The best way to learn is to observe about the worlds we live in. There has not being a research that describes a normal or abnormal method of problem recipe or solving. As a researcher, I investigate the methods of problem recipe in both natural and universal worlds. The natural world is based solely on the activities of the hand and without it, one cannot hold or grasp a thing to know about. The universal worlds needs a lot of observation in order to know- what, when, how, who and more. There is a world full of air and hands waving in air is nothing to describe. The eyes can see to describe more and the body can feel to sense and observe much. The biological research methods of the senses is what we can observe to describe as reactions to the physical world around us. There are four main principles of scientific study namely: *empiricism, determinism, parsimony* and *testability*. The first

principle is empiricism. This is about observations. The empirical nature of science will name influential people like Galileo, Plato and Aristotle. These people used observation ideas to study about the world in their works. Copernicus and Newton are contemporaries in the use of observation in scientific work.

Bibliography

[Dawn]Dawn M. McBride. The Process of Research in Psychology, Sage Publications, 2010.

[UML] Unified Modeling Language. UML Specification http://schema.omg.org/spec/UML/1.3 (Accessed 2017).

[JAVA] JAVA, Programming Language, http://www.oracle.com/jdk (Accessed 2018).

[UMBRELLO] Umbrello UML Modeller, UML Modeling http://uml.sf.net

[APPIAH]An Algorithmic Beauty in Problem Recipe, Patent Study, Department of Ghana, Patent Registry Office, 2016.

[APPIAH-REM]Application of Golden-Magic Solution, Patent Study, Department of Ghana, Patent Registry Office, 2018.

Appendix A: Computer Codes

Symbolic Solution Code

Computer Code:

```
public class SymbolicSoln
{
private String partySolver;
private String recipe;
private String nature;

public enum recipeType
    { GOLDEN(“golden”), RATIONAL(“rational”),
                MAGIC(“magic”)
   };

public enum solverType
   {
    POSITIVE(“positiveSlr”),
    NEGATIVE(“negativeSlr”)
   };

 public String getPartySolver(int solver){
   switch(solver)
    {
        case 1:   partySolver = solverType.
           POSITIVE.toString();
      case 2:   partySolver = solverType.
           NEGATIVE.toString();
    }
return partySolver;
}

public String getProblemRecipe(int recipe){
         switch(recipe)
         {
       case 1:   this.recipe = recipeType.
             GOLDEN.toString();
       case 2:   this.recipe = recipeType.
           RATIONAL.toString();
```

```
        case 3:this.recipe = recipeType.
                MAGIC.toString();
              }
           return this.recipe;
     }
    public void setPartySolver(int solver){
      switch(solver)
       {
           case 1:   partySolver = solverType.
              POSITIVE.toString();
        case 2:   partySolver = solverType.
              NEGATIVE.toString();
       }
  }
public void setProblemRecipe(int recipe){
           switch(recipe)
           {
        case 1:   this.recipe = recipeType.
               GOLDEN.toString();
        case 2:   this.recipe = recipeType.
           RATIONAL.toString();
       case 3: this.recipe = recipeType.
               MAGIC.toString();
           }
  }

  public String getNature(){
  return nature;
  }

  public void setNature(String nature){
  this.nature=nature;
      }
  }
```

Computer Code: ProblemSoln

```
public class ProblemSoln
{
private String problemType;
private String solutionType;

public enum pemType
```

```
        {
         NATURAL("natural"),
         UNIVERSAL("universal")
        };

    public enum slnType
        {
         SYMBOLIC("symbolic"),
         GOLDEN_MAGIC("golden-magic")
        };

    public String getProblemType(int proType)
    { switch(proType) //0 return default value
              {
         case 1:   this.problemType = pemType.
                NATURAL.toString();
        case 2:     this.problemType = pemType.
             UNIVERSAL.toString();
              }
    return this.problemType;
    }

    public String getSolutionType(int solType)
    { switch(solType) //0 return default value
              {
         case 1:   this.solutionType = slnType.
                SYMBOLIC.toString();
       case 2:     this.solutionType = slnType.
            GOLDEN_MAGIC.toString();
              }
     return this.solutionType;
    }

    public void setProblemType(int proType)
    {
       switch(proType)
              {
         case 1:   this.problemType = pemType.
                NATURAL.toString();
        case 2:     this.problemType = pemType.
             UNIVERSAL.toString();
              }
     }
```

```
public void setSolutionType(int solType)
{  switch(solType)
          {
     case 1:   this.solutionType = slnType.
             SYMBOLIC.toString();
   case 2:     this.solutionType = slnType.
       GOLDEN_MAGIC.toString();
          }
 }
```

Computer Code: Decomposition

```
public class Decomposition
{
private SymbolicSoln symbol;
private ProblemSoln problem;
private String settlement;
private int settleCount;

private enum settleType{
SETTLE_UP("settle_up"),
SETTLE_DOWN("settle_down")
};

public int getSettleCount(){
return this.settleCount;
}

public void setSettleCount(int count){
this.settleCount=count;
}

public SymbolicSoln getSymbol(){
return this.symbol=soln;
     }

public ProblemSoln getProblem(){
return this.problem;
   }

public void setSymbol(SymbolicSoln soln){
this.symbol=soln;
```

```
        }

    public void setProblem(ProblemSoln soln){
     this.problem=soln;
       }

    public void setSettlement(settleType type){
    this.settlement=type.toString();
      }

    public settleType getSettlement()  {
    this.settlement;
      }
    }
```

Appendix B: UML Metamodels in Symbiolic Decomposition.